

Amendments to the Claims:

This listing of claims will replace all prior versions of claims in the present application:

Listing of Claims:

1-15. (Canceled)

16. (Currently amended) A method for supporting a plurality of graphical user interface (GUI) application programming interfaces (APIs), the method being performed by a computer, the method comprising:

translating a plurality of elements of a query model into objects that are independent of any type of data structure associated with the plurality of GUI APIs, the plurality of elements being translated through use of a model content provider in communication with the query model, the plurality of elements representing a database statement;

passing the translated objects from the model content provider to a first content viewer in communication with the model content provider, the first content viewer supporting multiple GUI APIs;

passing the translated objects from the first content viewer to a second content viewer, the second content viewer being in communication with the first content viewer and [[an]]a GUI application written to run on a specific GUI API of the plurality of GUI APIs, wherein the first and second content viewers interface with the model content

~~provider within a GUI-model infrastructure, wherein each of the first and second content viewers [[is]] are defined by a hierarchical set of classes, wherein higher levels of the hierarchical set of classes the first content viewer is a higher level, comprise more abstract structures non-GUI specific structure that is created without regard to which GUI API will be used, and wherein the second content viewer is a lower levels level of the hierarchical set of classes comprise more GUI-specific structures structure that are is created specifically to the GUI API with which they communicate it communicates; and~~
 using the second content viewer to manipulate the translated objects into one or more types of data structures required by the specific GUI API for use by the application.

17. (Previously Presented) The method of claim 16, wherein the one or more types of data structures comprise tables, trees, or lists.

18. (Previously Presented) The method of claim 16, wherein the database statement is a structured query language (SQL) statement.

19. (Currently amended) The method of claim 16, further comprising:
 receiving information from the application via the first content viewer and the second content viewer at the model content provider, the received information being independent of any type of data structure, wherein the information indicates [[on]]one of an addition, a modification, and a deletion of at least one element in the query model; and

creating one or more additional elements using the model content provider based on the received information responsive to the received information being an addition to the plurality of elements in the query model.

20. (Previously presented) The method of claim 16, further comprising:

receiving information from the application via the first content viewer and the second content viewer at the model content provider, the received information being independent of any type of data structure; and

removing one or more of the plurality of elements from the query model using the model content provider responsive to the received information being a deletion of the one or more elements in the query model.

21. (Previously Presented) The method of claim 16, further comprising:

providing both data and image information for each of the plurality of elements in the query model to the first content viewer using the model content provider.

22-28. (Canceled)

29. (Previously Presented) The method of claim 19, further comprising linking the one or more additional elements in the query model.

30. (Currently Amended) The method of claim ~~[[19]]~~16, wherein the ~~one or more additional elements comprises a table~~first content viewer comprises wrapper code for a

particular API, and wherein a new GUI application using a different GUI API is enabled to write its own wrapper code and reuse the second content viewer, which is non-GUI specific.

31. (New) A model content provider for supporting a plurality of graphical user interface (GUI) application programming interfaces (APIs), the model content provider being implemented by a computer, the model content provider operable to:

translate a plurality of elements of a query model into objects that are independent of any type of data structure associated with the plurality of GUI APIs, the plurality of elements being translated through use of the model content provider in communication with the query model, the plurality of elements representing a database statement;

pass the translated objects to a first content viewer in communication with the model content provider, the first content viewer supporting multiple GUI APIs;

pass the translated objects from the first content viewer to a second content viewer, the second content viewer being in communication with the first content viewer and a GUI application written to run on a specific GUI API of the plurality of GUI APIs, wherein the first and second content viewers interface with the model content provider within a GUI-model infrastructure, wherein the first and second content viewers are defined by a hierarchical set of classes, wherein the first content viewer is a higher level, non-GUI specific structure that is created without regard to which GUI API will be used, wherein the second content viewer is a lower level GUI-specific structure that is created specifically to the GUI API with which it communicates; and

use the second content viewer to manipulate the translated objects into one or more types of data structures required by the specific GUI API for use by the application.

32. (New) An article of manufacture comprising a computer program carrier readable by a computer and embodying one or more instructions executable by the computer for supporting a plurality of graphical user interface (GUI) application programming interfaces (APIs), the instructions comprising:

program instructions for translating a plurality of elements of a query model into objects that are independent of any type of data structure associated with the plurality of GUI APIs, the plurality of elements being translated through use of the model content provider in communication with the query model, the plurality of elements representing a database statement;

program instructions for passing the translated objects to a first content viewer in communication with the model content provider, the first content viewer supporting multiple GUI APIs;

program instructions for passing the translated objects from the first content viewer to a second content viewer, the second content viewer being in communication with the first content viewer and a GUI application written to run on a specific GUI API of the plurality of GUI APIs, wherein the first and second content viewers interface with the model content provider within a GUI-model infrastructure, wherein the first and second content viewers are defined by a hierarchical set of classes, wherein the first content viewer is a higher level, non-GUI specific structure that is created without regard to which GUI API will be used, wherein the second content viewer is a lower level GUI-

specific structure that is created specifically to the GUI API with which it communicates;

and

program instructions for using the second content viewer to manipulate the translated objects into one or more types of data structures required by the specific GUI API for use by the application.